



Implémentez un modèle de scoring

Preparé par: Sidi Teyib BEDDY EL MOUSTAPHA

Mentor : El Hadji Abdoulaye Thiam

Évaluateur: Aurelien Quillet

La Problématique

- ❖ Comment développer un outil de "scoring crédit" basé sur un algorithme de classification en utilisant des sources de données variées, afin de calculer la probabilité de remboursement d'un crédit et de classer les demandes en crédit accordé ou refusé, tout en répondant à la demande croissante de transparence des clients vis-à-vis des décisions d'octroi de crédit ?
- ❖ Comment concevoir un dashboard interactif pour les chargés de relation client, permettant d'expliquer de manière transparente les décisions d'octroi de crédit tout en offrant aux clients un accès facile à leurs informations personnelles pour exploration ?

PLAN

1. Introduction
2. Présentation du jeu de donnée
3. Présentation de la modélisation
4. Visualisation du tracking via MLFlow UI
5. Présentation du pipeline de déploiement
6. Présentation de l'analyse de data drift
7. Présentation et démo du dashboard déployé sur le Cloud
8. Conclusion

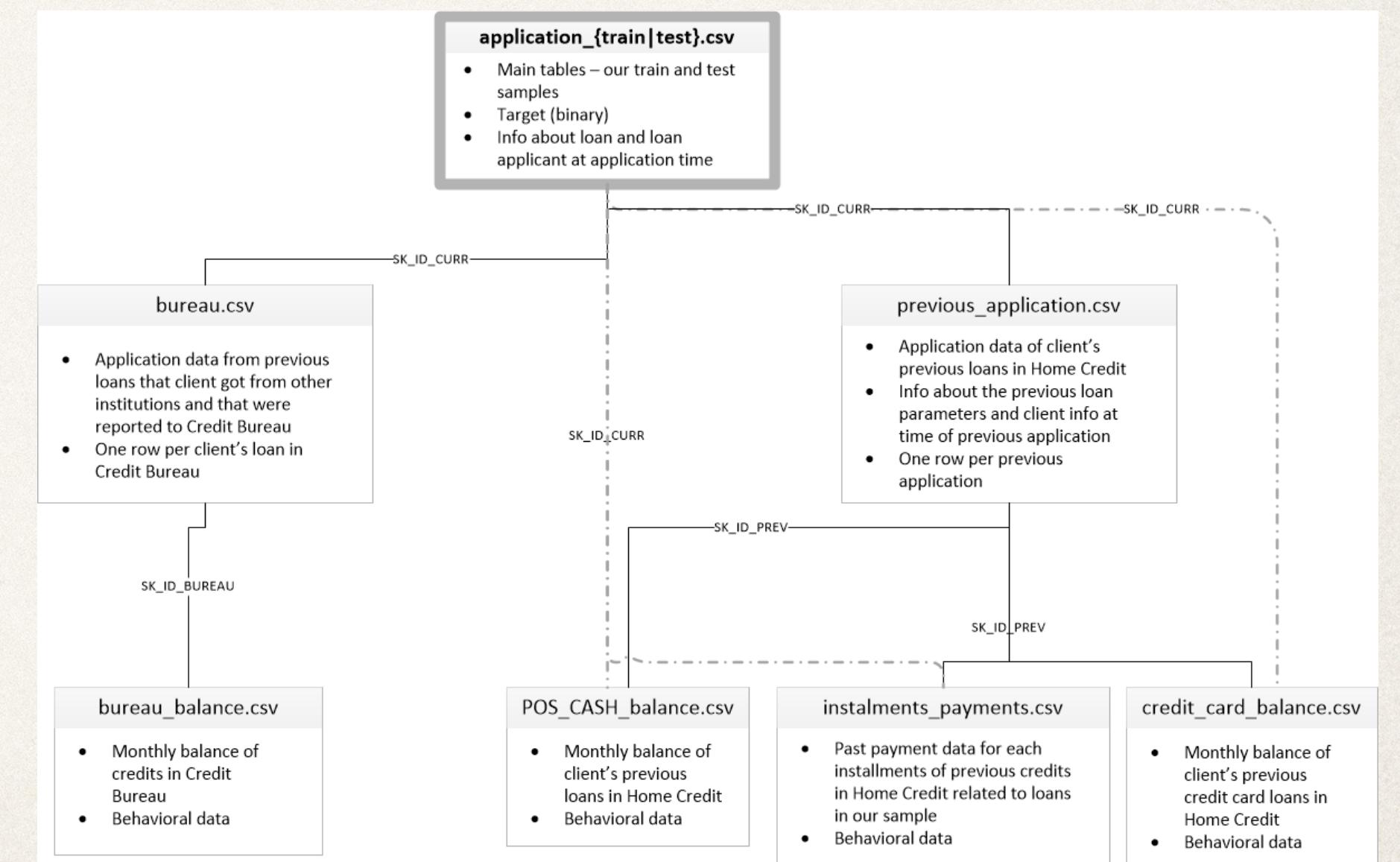
Introduction

Le projet vise à développer un modèle de scoring crédit pour évaluer la probabilité de défaut de paiement des clients de "Prêt à Dépenser". L'objectif est de créer un modèle robuste et interprétable, tout en mettant l'accent sur la transparence des décisions d'octroi de crédit. Un dashboard interactif sera également mis en place pour fournir des explications claires aux clients sur les critères de leur évaluation de crédit. Une analyse de data drift sera réalisée pour évaluer la stabilité des données dans le temps. Le but ultime est d'offrir un service de qualité tout en minimisant les risques financiers liés aux prêts accordés.

Présentation du jeu de donnée

Le jeu de données d'entraînement, disponible au format CSV sur Kaggle, comprend 307 000 observations, chacune étant décrite par 121 caractéristiques telles que l'âge, le sexe, l'emploi, le logement, les revenus, les informations de crédit, la notation externe, etc. En plus de ce fichier principal, nous disposons également de 6 autres fichiers de données au format CSV, ainsi qu'un fichier de description qui ont été chargés dans le notebook et rapidement examinés pour identifier le type d'informations disponibles.

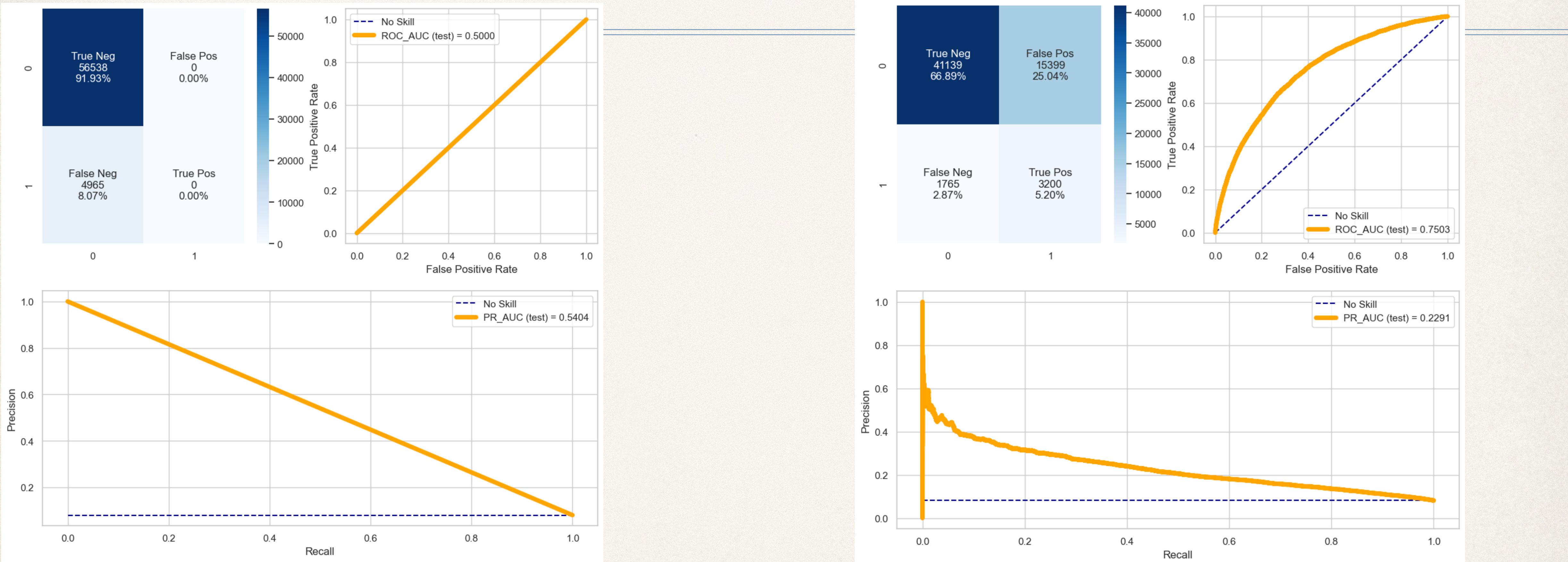
Cette image montre les colonnes communes entre les fichiers, qui serviront de clé de jointure pour fusionner les données avec les démarches entreprises.



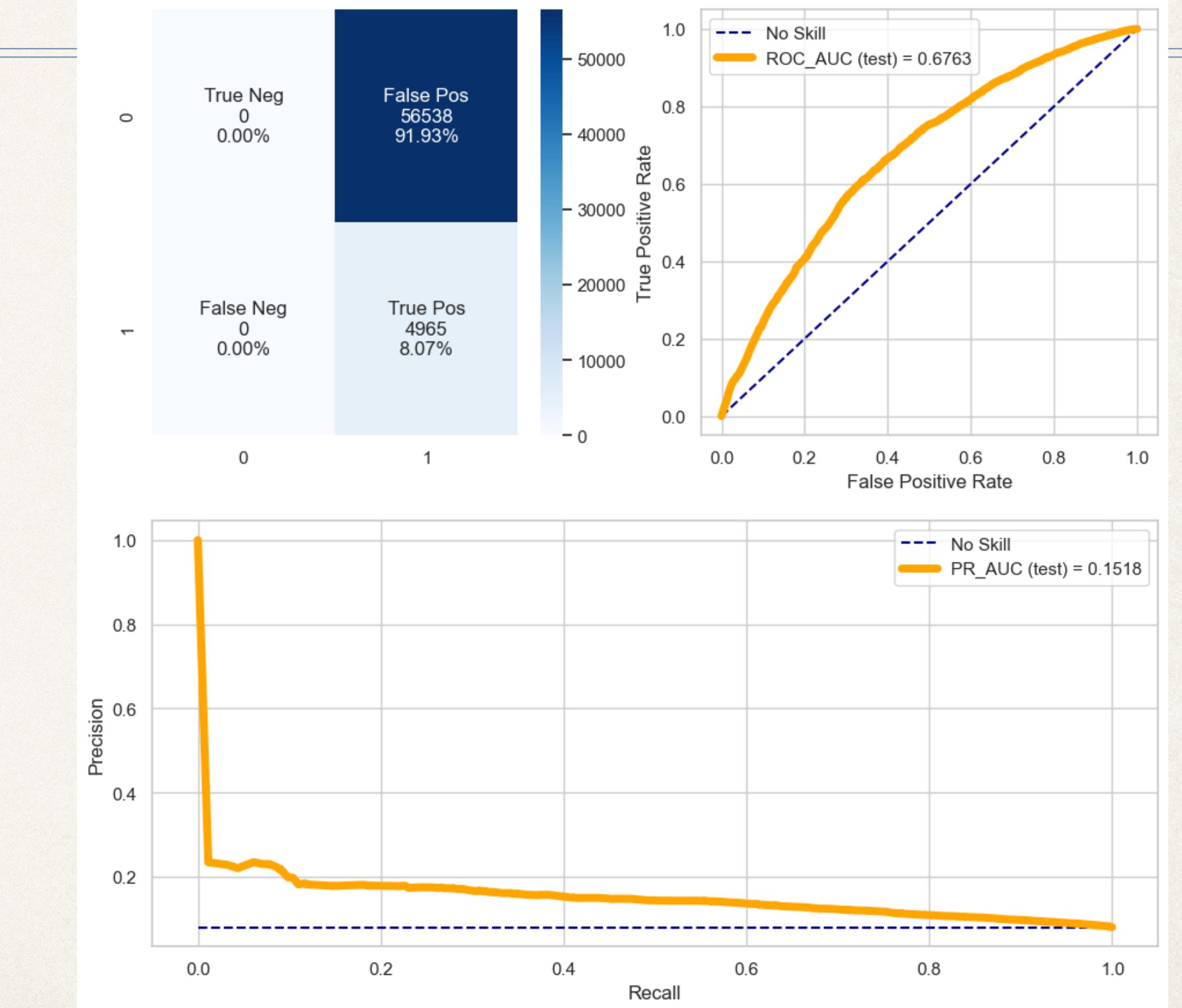
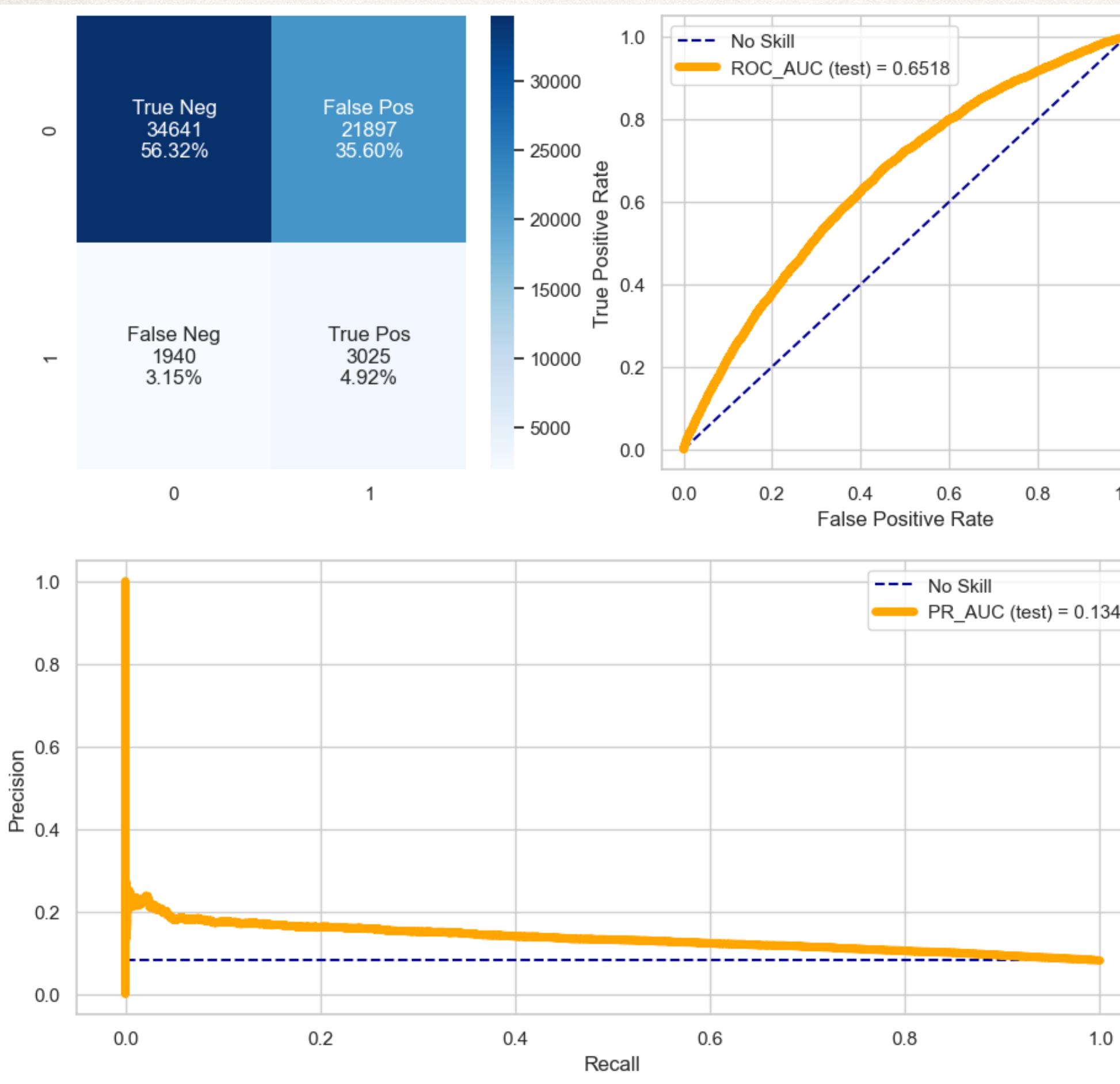
Analyse Exploratoire des Données



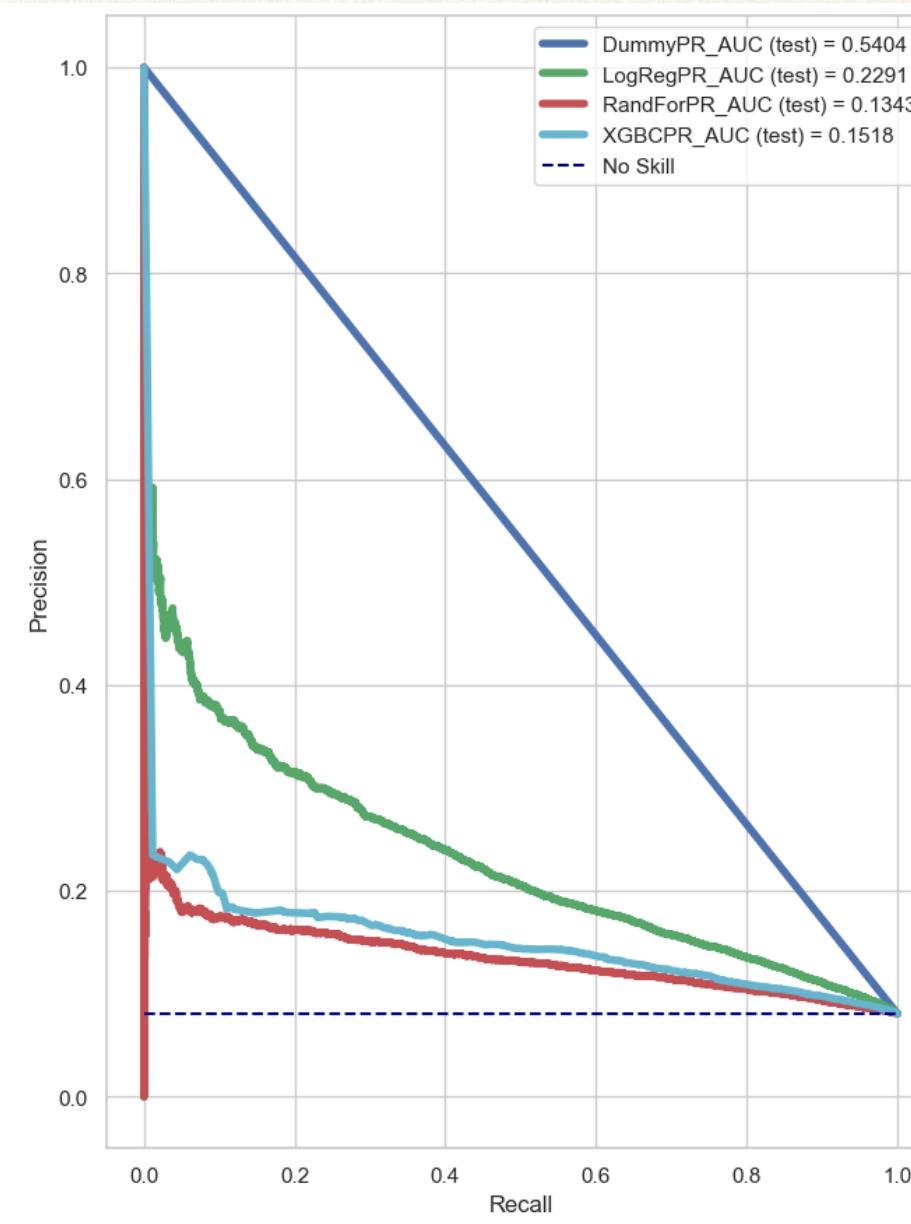
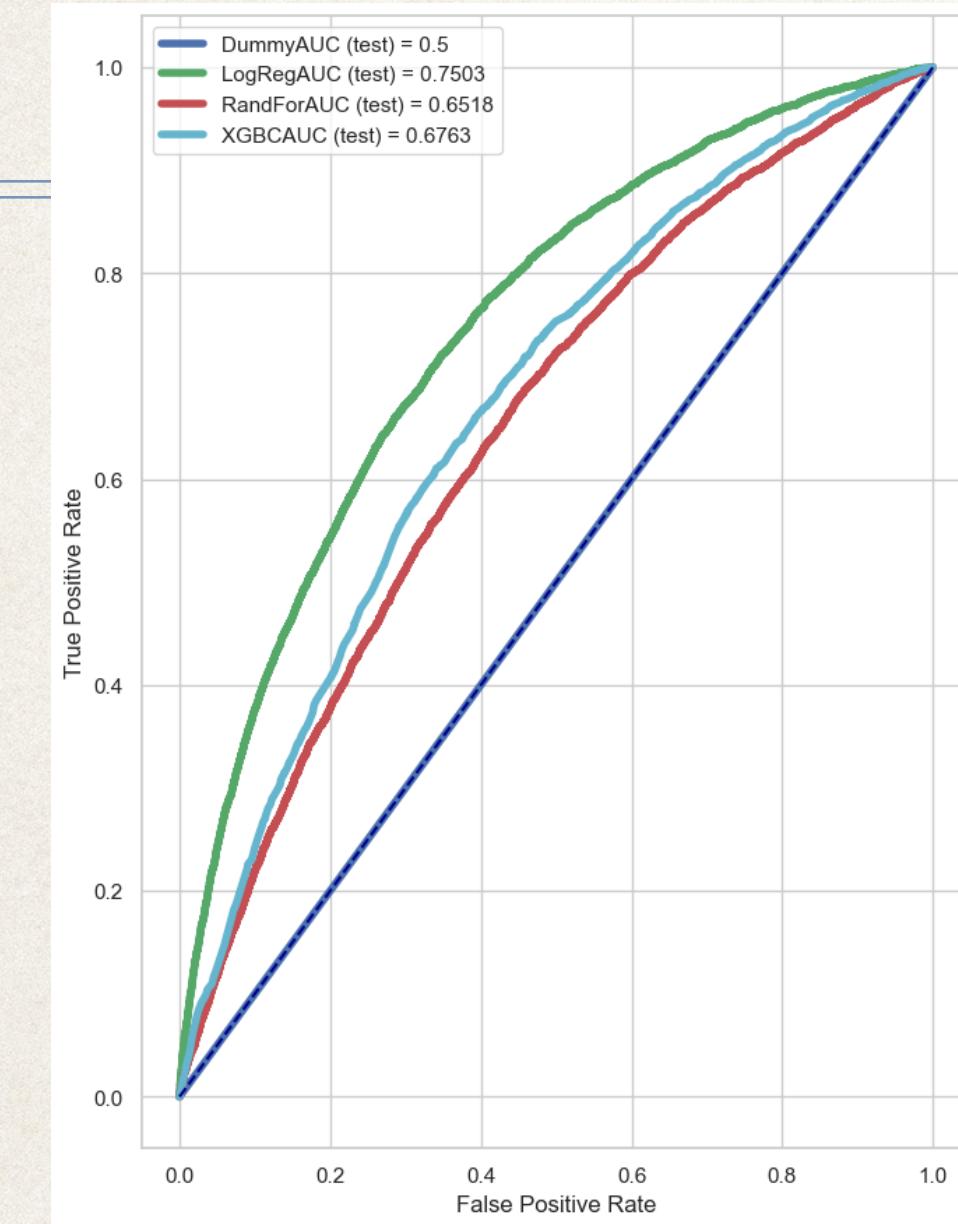
Présentation de la modélisation



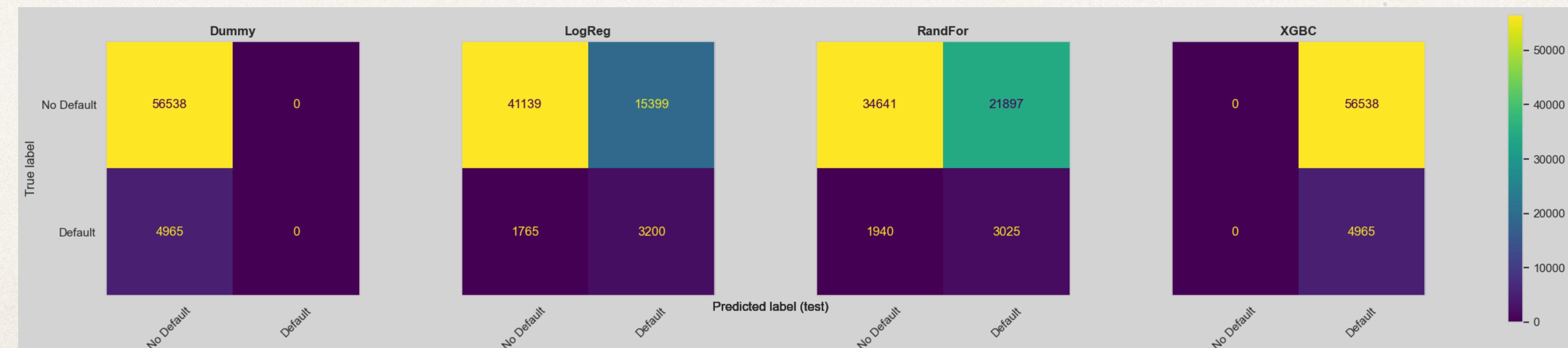
Présentation de la modélisation



Présentation de la modélisation



model_scores					
	Data	Model	Custom_score	ROC_AUC	PR_AUC
0	Train	Dummy		0.8073	0.5 0.540365
1	Test	Dummy		0.8073	0.5 0.540364
2	Train	LogReg		0.5384	0.7508 0.223985
3	Test	LogReg		0.5374	0.7503 0.229102
4	Train	RandFor		0.561	0.7397 0.184874
5	Test	RandFor		0.6715	0.6518 0.134322
6	Train	XGBC		0.561	0.7397 0.184874
7	Test	XGBC		0.9193	0.6763 0.151798



FEATURES IMPORTANTS

```

import joblib
import pandas as pd
import matplotlib.pyplot as plt

# Charger le pipeline à partir du fichier enregistré
pipeline = joblib.load('Models/best_model_XGBC.joblib')

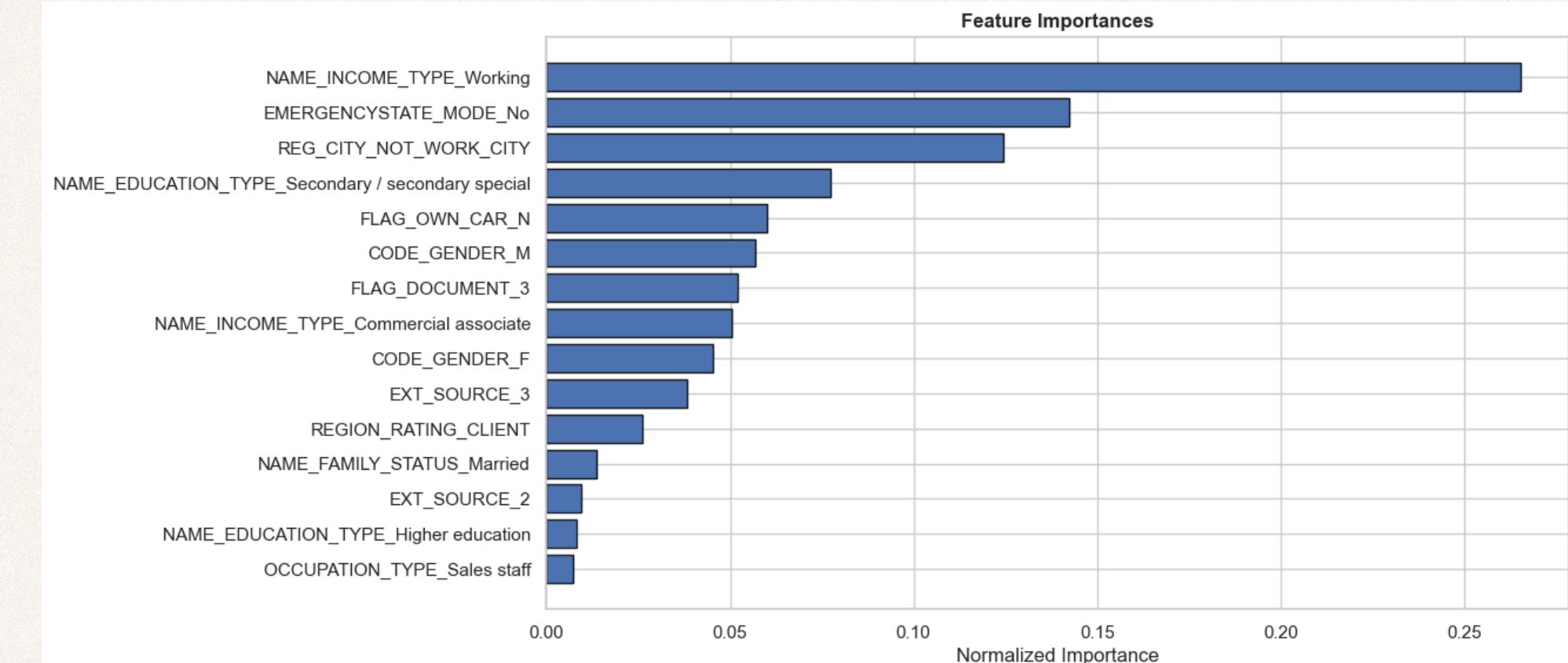
# Accéder au modèle dans le pipeline
model = pipeline.named_steps['classifier']

# Accéder aux caractéristiques importantes du modèle
feature_importances = model.feature_importances_

# Feature importances
feat_name = train_set.drop(['SK_ID_CURR', 'TARGET'], axis=1).columns
feat_valu = feature_importances[:len(feat_name)] # Ajustement de la longueur

feat_frame = pd.DataFrame()
feat_frame['feature'] = feat_name
feat_frame['importance'] = feat_valu

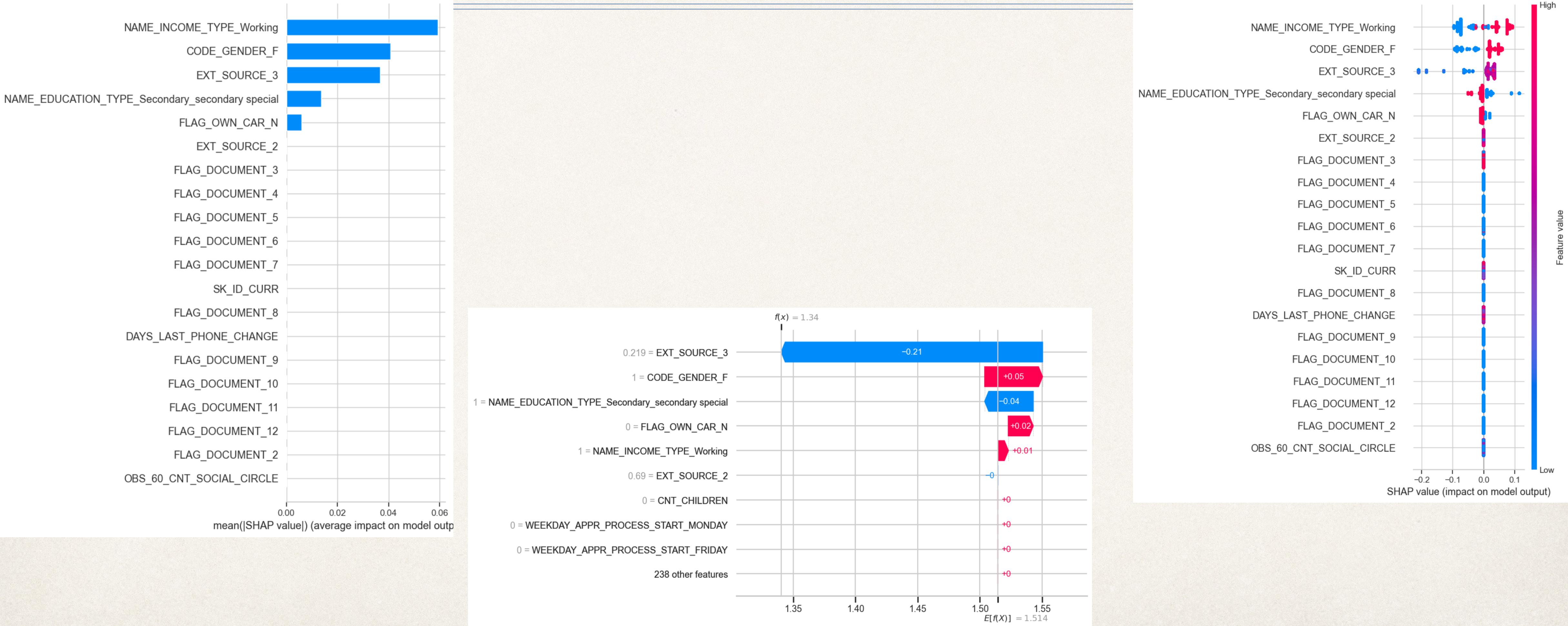
```



index	feature	importance	importance_normalized
0	130	NAME_INCOME_TYPE_Working	0.265235
1	244	EMERGENCYSTATE_MODE_No	0.142280
2	25	REG_CITY_NOT_WORK_CITY	0.124304
3	135	NAME_EDUCATION_TYPE_Secondary / secondary special	0.077326
4	113	FLAG_OWN_CAR_N	0.060222
...
241	90	FLAG_DOCUMENT_14	0.000000
242	91	FLAG_DOCUMENT_15	0.000000
243	92	FLAG_DOCUMENT_16	0.000000
244	93	FLAG_DOCUMENT_17	0.000000
245	123	NAME_TYPE_SUITE_Unaccompanied	0.000000

246 rows × 4 columns

FEATURES IMPORTANTS



Visualisation du tracking via MLFlow UI

The screenshot displays the MLflow UI interface. On the left, the 'Experiments' page shows a list of experiments, with 'PROJET7' selected. The main area shows a table of runs with columns: Run Name, Created, Dataset, Duration, Source, and Mode. The table lists four runs: XGBoostClassifier, LogisticRegression, RandomForestClassifier, and DummyClassifier_baseline, all created 17-19 minutes ago. On the right, the 'Runs' page provides detailed information for a specific run. It includes the Git Commit (bc73a46ceee2ee6881386ce07dd09c678748f9ad), User (sbeddy), Status (FINISHED), Lifecycle Stage (active), and a Metrics section with six entries: accuracy (0.922), f1_score (0.071), log_loss (2.829), precision (0.588), recall (0.038), and roc_auc (0.518).

MLflow 2.5.0 Experiments Models GitHub Docs

PROJET7 Provide Feedback

Search Experiments

Default PROJET7

Table view Chart view Artifact view metrics.rmse < 1 and params.model = "tree" Time created Refresh

State Active Sort: Created Columns Expand rows

Run Name	Created	Dataset	Duration	Source	Mode
XGBoostClassifier	17 minutes ago	-	49.7s	test1.py	skl
LogisticRegression	17 minutes ago	-	3.0s	test1.py	skl
RandomForestClassifier	19 minutes ago	-	1.3min	test1.py	skl
DummyClassifier_baseline	19 minutes ago	-	1.6s	test1.py	skl

Git Commit: bc73a46ceee2ee6881386ce07dd09c678748f9ad User: sbeddy Status: FINISHED Lifecycle Stage: active

Metrics (6)

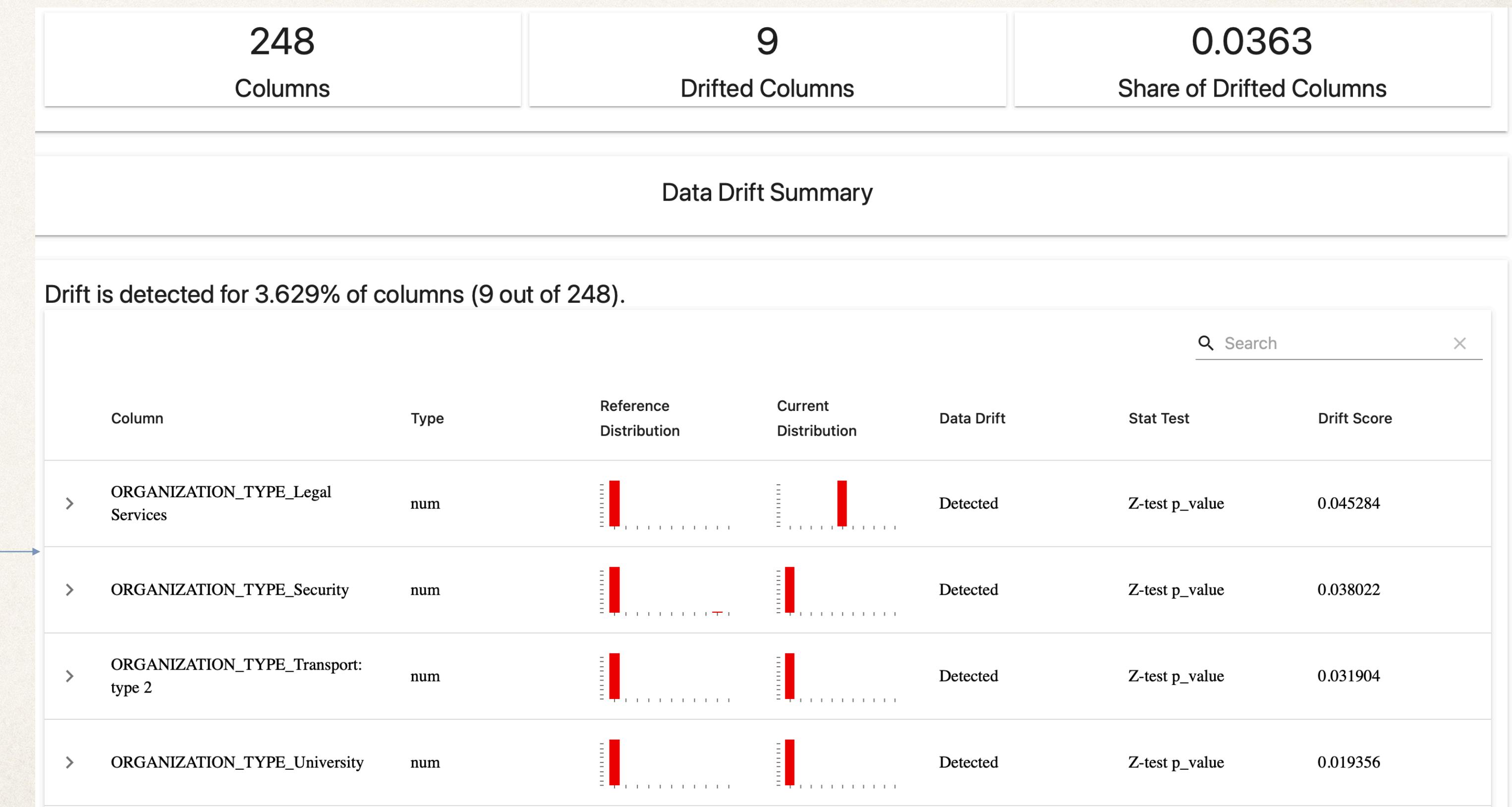
Name	Value
accuracy	0.922
f1_score	0.071
log_loss	2.829
precision	0.588
recall	0.038
roc_auc	0.518

Présentation de l'analyse de data drift

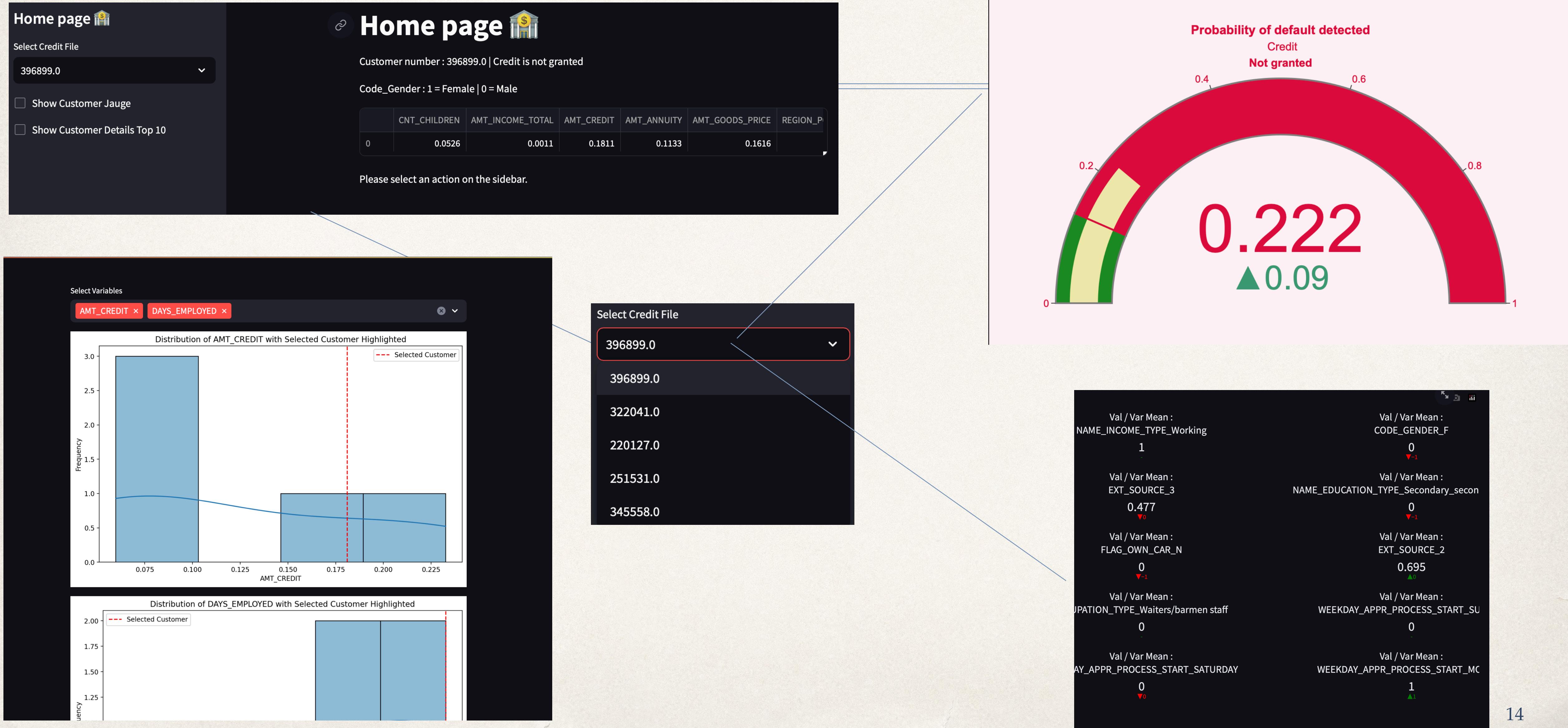
```
from evidently.test_suite import TestSuite
from evidently.test_preset import DataStabilityTestPreset

from evidently.report import Report
from evidently.metric_preset import DataDriftPreset

data_drift_report = Report(metrics=[DataDriftPreset()],
                           )
data_drift_report.run(current_data=pred_data, reference_data=train_data, column_mapping=None)
data_drift_report
```



Présentation et démo du dashboard déployé sur le Cloud



Conclusion

- ✿ En conclusion de ce projet "Implémentez un modèle de scoring", nous avons réussi à développer un outil performant et transparent pour évaluer la probabilité qu'un client rembourse son crédit. Grâce à l'analyse de données comportementales et à l'entraînement de plusieurs modèles, nous avons pu sélectionner le modèle le plus optimal en minimisant le coût métier tout en obtenant les meilleures aires sous les courbes ROC_AUC et PR_AUC.
- ✿ Le dashboard interactif que nous avons créé permet aux chargés de relation client d'expliquer de manière claire et transparente les décisions d'octroi de crédit aux clients. De plus, les clients peuvent accéder à leurs informations personnelles et les explorer facilement, répondant ainsi à leur demande croissante de transparence.
- ✿ L'analyse du data drift nous a permis d'évaluer la stabilité et la cohérence des données utilisées par nos modèles, ce qui nous a aidés à maintenir la fiabilité et la pertinence de notre outil au fil du temps.

Dans l'ensemble, ce projet a été un succès en fournissant un outil de scoring crédit efficace et compréhensible, répondant aux besoins des professionnels de la banque et des clients, tout en garantissant la transparence et la qualité des décisions d'octroi de crédit.